

# Module 7: Introduction to Gibbs Sampling

Yutong Shao

2023-03-06

## Introduction

In this lab, we will derive conditional distributions, code a Gibbs sampler, and analyze the output of the Gibbs sampler.

Consider the following Exponential model for observation(s)  $x_{1:n} = (x_1, \dots, x_n)$ .<sup>1</sup>:

$$p(x|a, b) = ab \exp(-abx) \mathbf{1}_{x>0},$$

where the  $x_i$  are assumed to be iid for  $i = 1, \dots, n$ . and suppose the prior is

$$p(a, b) = \exp(-a - b) \mathbf{1}_{a,b>0}.$$

We want to sample from the posterior  $p(a, b|x_{1:n})$ .

Tasks

1. Find the conditional distributions needed for implementing a Gibbs sampler.
2. Code up our own Gibbs sampler using part (1).
3. Run the Gibbs sampler, providing convergence diagnostics.
4. Plot a histogram or a density estimate of the estimated posterior using (2) and (3).
5. How do you know that your estimated posterior in (3) is reliable?

## Task 1: Conditional distributions

Consider the following Exponential model for observation(s)  $x_{1:n} = (x_1, \dots, x_n)$ :

$$p(x|a, b) = ab \exp(-abx) \mathbf{1}_{x>0}$$

and suppose the prior is

$$p(a, b) = \exp(-a - b) \mathbf{1}_{a,b>0}.$$

Our final goal is to sample from the posterior  $p(a, b|x)$ .

$$\begin{aligned} p(\mathbf{x}|a, b) &= \prod_{i=1}^n p(x_i|a, b) \\ &= \prod_{i=1}^n ab \exp(-abx_i) \\ &= (ab)^n \exp\left(-ab \sum_{i=1}^n x_i\right). \end{aligned}$$

---

<sup>1</sup>Please note that in the attached data there are 30 observations, which can be found in data-exponential.csv.

The function is symmetric for  $a$  and  $b$ , so we only need to derive  $p(a|\mathbf{x}, b)$ .  
This conditional distribution satisfies

$$\begin{aligned}
 p(a|\mathbf{x}, b) &\propto_a p(a, b, \mathbf{x}) \\
 &= p(\mathbf{x}|a, b)p(a, b) \\
 &= (ab)^n \exp\left(-ab \sum_{i=1}^n x_i\right) \times \exp(-a - b)I(a, b > 0) \\
 &\propto_a a^n \exp(-abn\bar{x} - a)\mathbf{1}_{a>0} \\
 &= a^{n+1-1} \exp(-(bn\bar{x} + 1)a)\mathbf{1}_{a>0} \\
 &\propto_a \text{Ga}(a|n + 1, bn\bar{x} + 1).
 \end{aligned}$$

Therefore,  $p(a|b, x) = \text{Ga}(a|n + 1, bn\bar{x} + 1)$  and by symmetry,  $p(b|a, x) = \text{Ga}(b|n + 1, an\bar{x} + 1)$ .

We now load the packages we need and the observed data  $X$ .

```
library(MASS)
library(coda)
library(ggplot2)
X=c(
0.0371331907051745,0.271390503356533,2.40473359881563,
0.279129389456793,3.89335288168723,6.07299593757399,
2.0008345204379,0.374140078068601,0.269576448538397,
1.316028759158,2.42209106222679,1.12711988643488,
0.847399458693192,0.0916006407183253,3.15166571883227,
0.122355493965856,1.6631530013599,2.31930895969008,
0.254301124443481,2.91553389072551,1.99536939381821,
2.46937213527566,9.4394789881428,5.27195433646316,
0.45310761063482,1.65998358370856,2.95869986327957,
2.82643679251663,0.407467252515871,1.07017463071046
)
```

## Task 2: Gibbs sampling code

Here the pseudo-code

```
#####
# This function is a Gibbs sampler
#
# Input
# a0: initial value for a
# b0: initial value for b
# S: number of iterations
# X: observed data, just a vector
#
# Output
# A two column matrix AB with samples
#   # for a in first column and
#   # for b in second column
#####
```

And here the code

```

sampleGibbs=function(a0, b0, S, X){
  # get sum, which is sufficient statistic
  x=sum(X)
  # get n
  n=length(X)
  # create empty matrix, allocate memory for efficiency
  AB=matrix(NA, nrow=S, ncol=2)
  # iterate the algorithm
  a=a0;b=b0
  for(j in 1:S){ # number of iteration
    AB[j,1]=a
    AB[j,2]=b
    a=rgamma(1,shape=n+1, rate=b*x+1)
    b=rgamma(1,shape=n+1, rate=a*x+1)
  }
  return(AB)
}

```

### Task 3: Run the Gibbs sampler

We start with some arbitrary initial values  $a_0 = b_0 = 1$ .

```

# run Gibbs sampler
a0=b0=1 # initial values = 1
S=10000
AB=sampleGibbs(a0 , b0 , S , X)
head(AB)

```

```

##           [,1]      [,2]
## [1,] 1.0000000 1.0000000
## [2,] 0.5092957 0.8105875
## [3,] 0.6787493 0.6567587
## [4,] 0.7442373 0.5887998
## [5,] 0.7659561 0.8472438
## [6,] 0.5198861 1.1419555

```

```

df=data.frame(s=1:S, a=AB[,1], b=AB[,2])

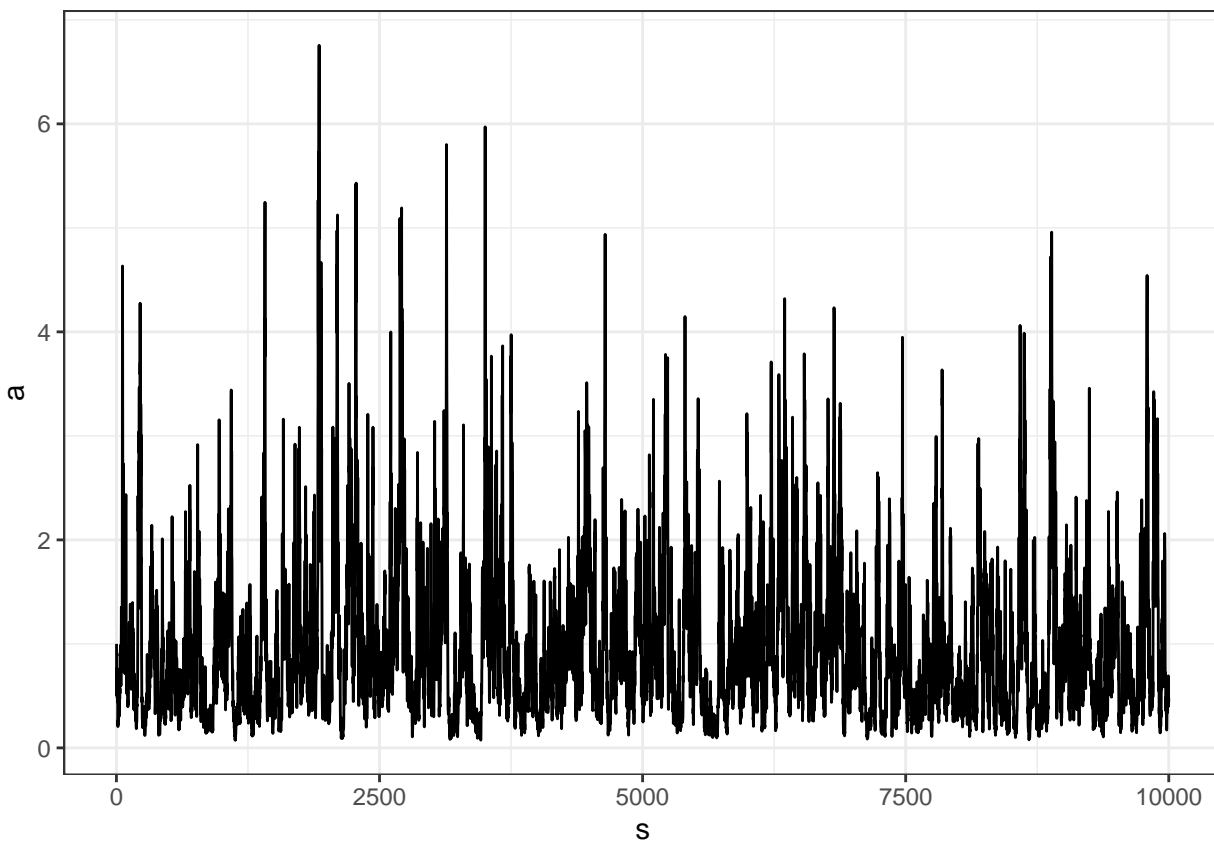
```

We now want to assess the convergence of the algorithm, so we use a traceplot

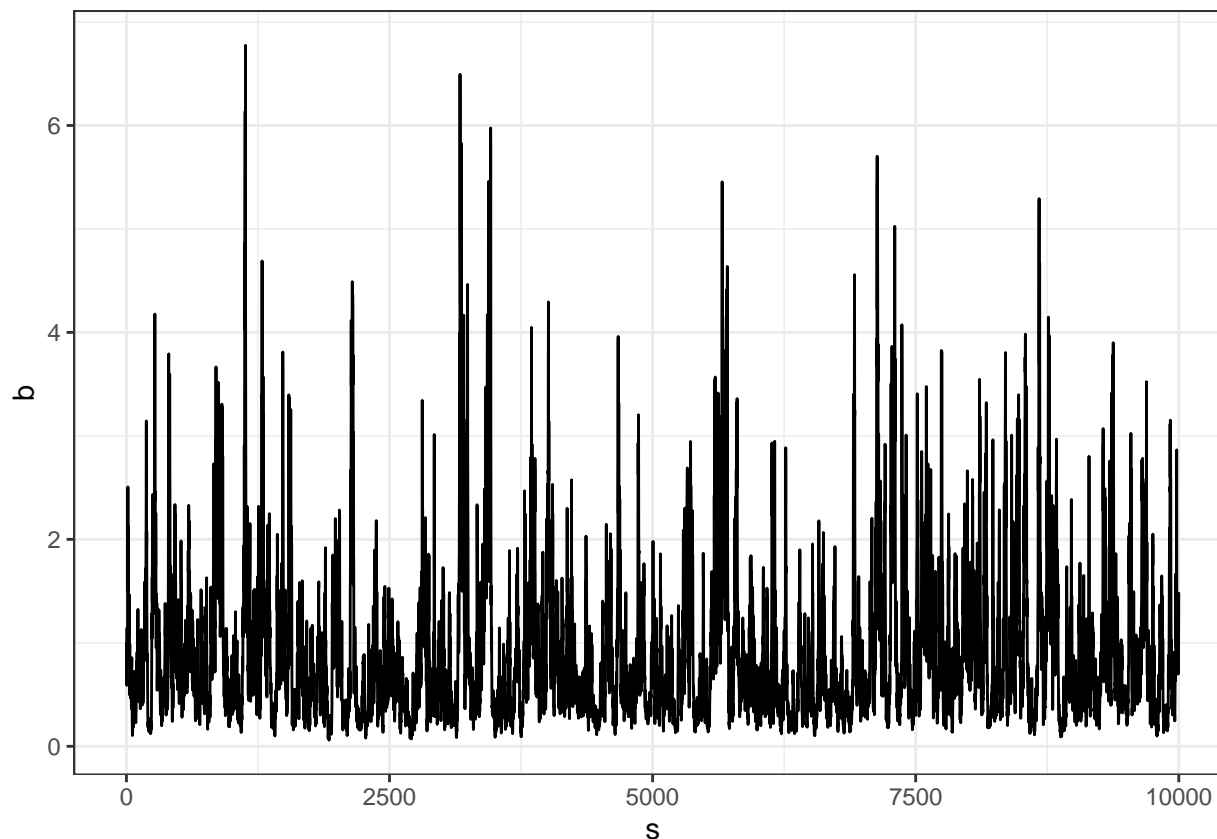
```

ggplot(df)+
  geom_line(aes(x=s, y=a))+
  theme_bw()

```



```
ggplot(df)+  
  geom_line(aes(x=s, y=b))+  
  theme_bw()
```



Let's also look at the Effective Sample Size ratio computing using the package coda.

```
effectiveSize(AB[,1])/S*100
```

```
##      var1
## 3.752515
```

```
effectiveSize(AB[,2])/S*100
```

```
##      var1
## 3.64166
```

This is quite low, and indeed, even from the traceplots we can see some **autocorrelation** (that is some local trend in the traces). [**Remark:** This is because the samples are too close to one another and are very likely to be correlated.]

A way to increase the ESS ratio, is through **thinning**. That is the process of **considering only samples at a certain distance from each other**

```
# AB is the original dataframe
# ABthinned is the thinned version
ABthinned=AB[seq(1,S,by=25),]
Sthinned=dim(ABthinned)[1]

print(Sthinned)
```

```
## [1] 400
```

```
effectiveSize(ABthinned[,1])/Sthinned*100
```

```
##      var1  
## 80.33565
```

```
effectiveSize(ABthinned[,2])/Sthinned*100
```

```
##      var1  
## 86.90805
```

In this case we have increased the informativity of our sample but we have also reduced its size. In a sense we have *compressed our sample*, because the ESS of both the original samples and the thinned version is similar even though we have a tenfold reduction in size.

```
c(effectiveSize(ABthinned[,1]),effectiveSize(AB[,1]))
```

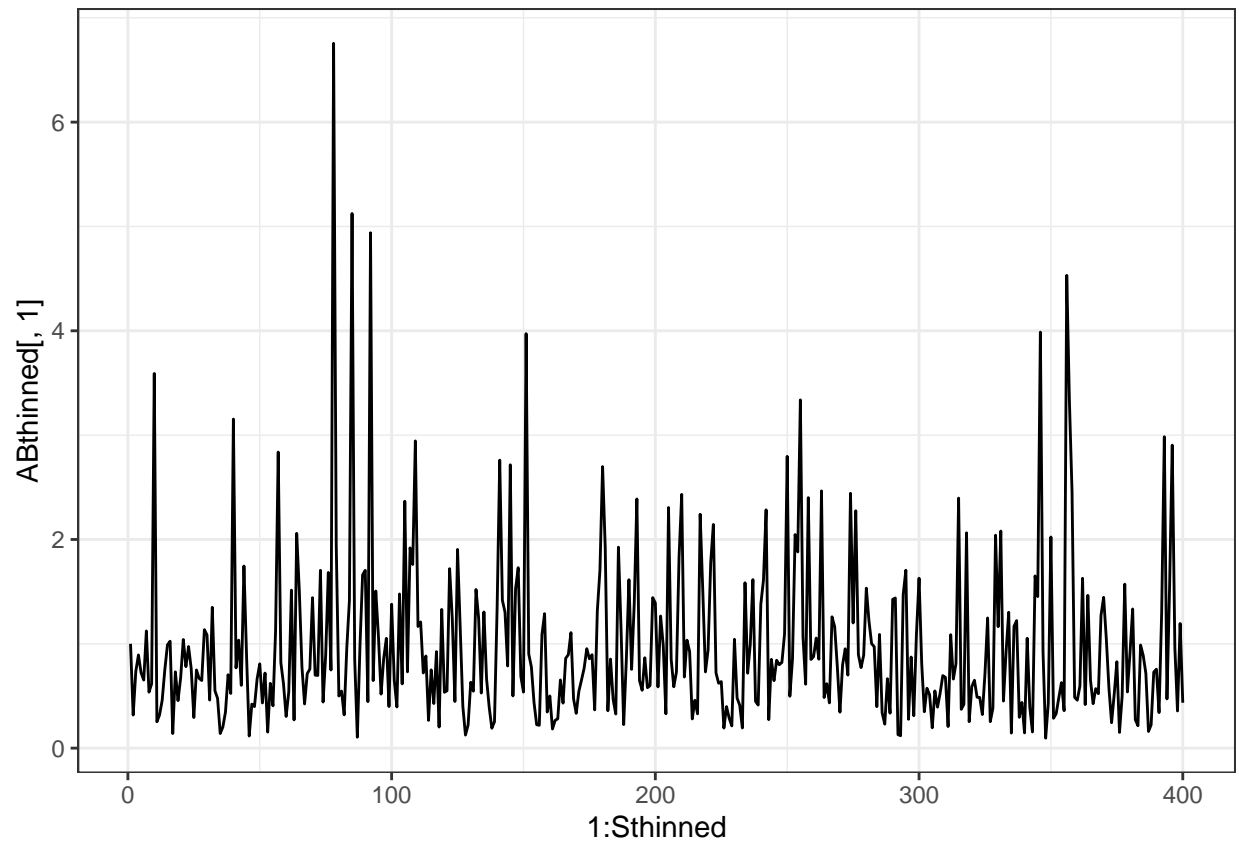
```
##      var1      var1  
## 321.3426 375.2515
```

```
c(effectiveSize(ABthinned[,2]),effectiveSize(AB[,2]))
```

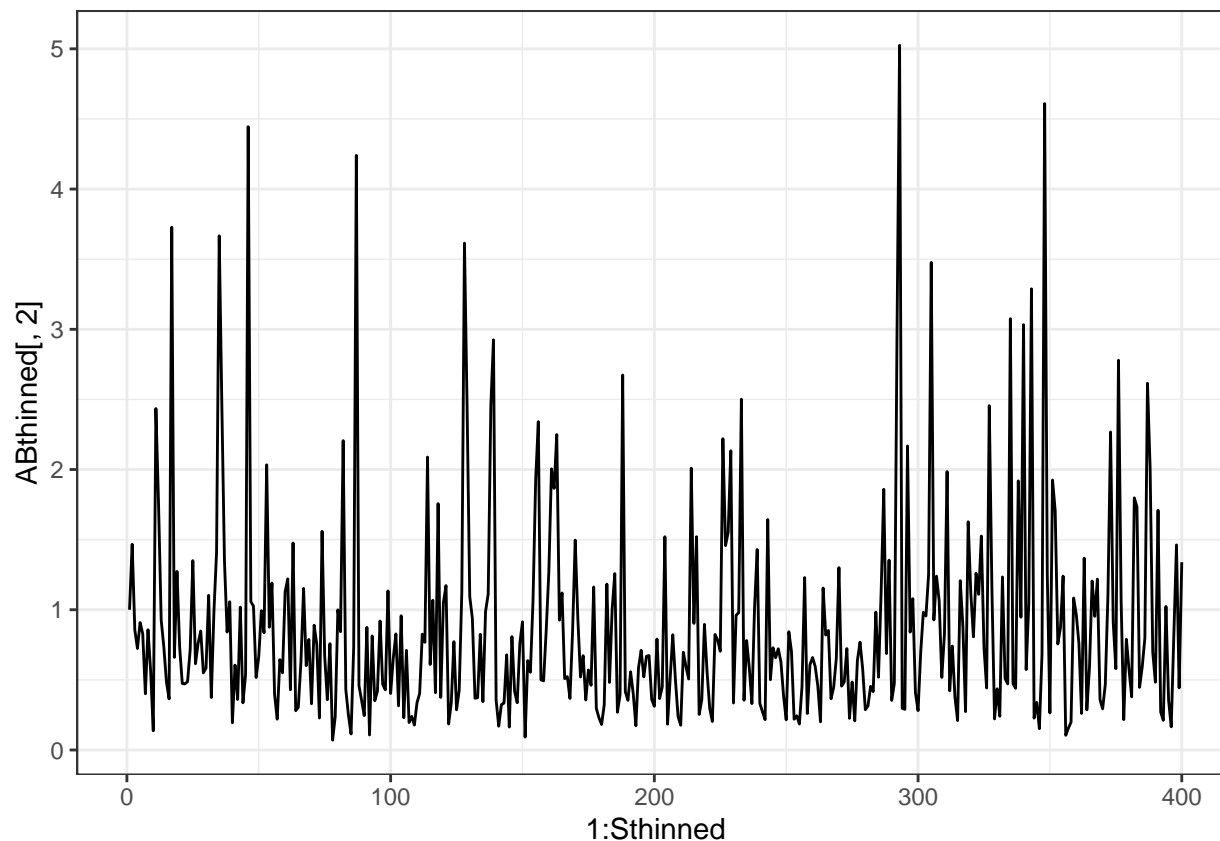
```
##      var1      var1  
## 347.6322 364.1660
```

Let's plot the traces of the thinned sample

```
ggplot()+  
  geom_line(aes(x=1:Sthinned, y=ABthinned[,1]))+  
  theme_bw()
```



```
ggplot()+  
  geom_line(aes(x=1:Sthinned, y=ABthinned[,2]))+  
  theme_bw()
```



These plots seems to indicate convergence to a stationary distribution. And though thinning we have manage to eliminate most of the correlation.

## Task 4

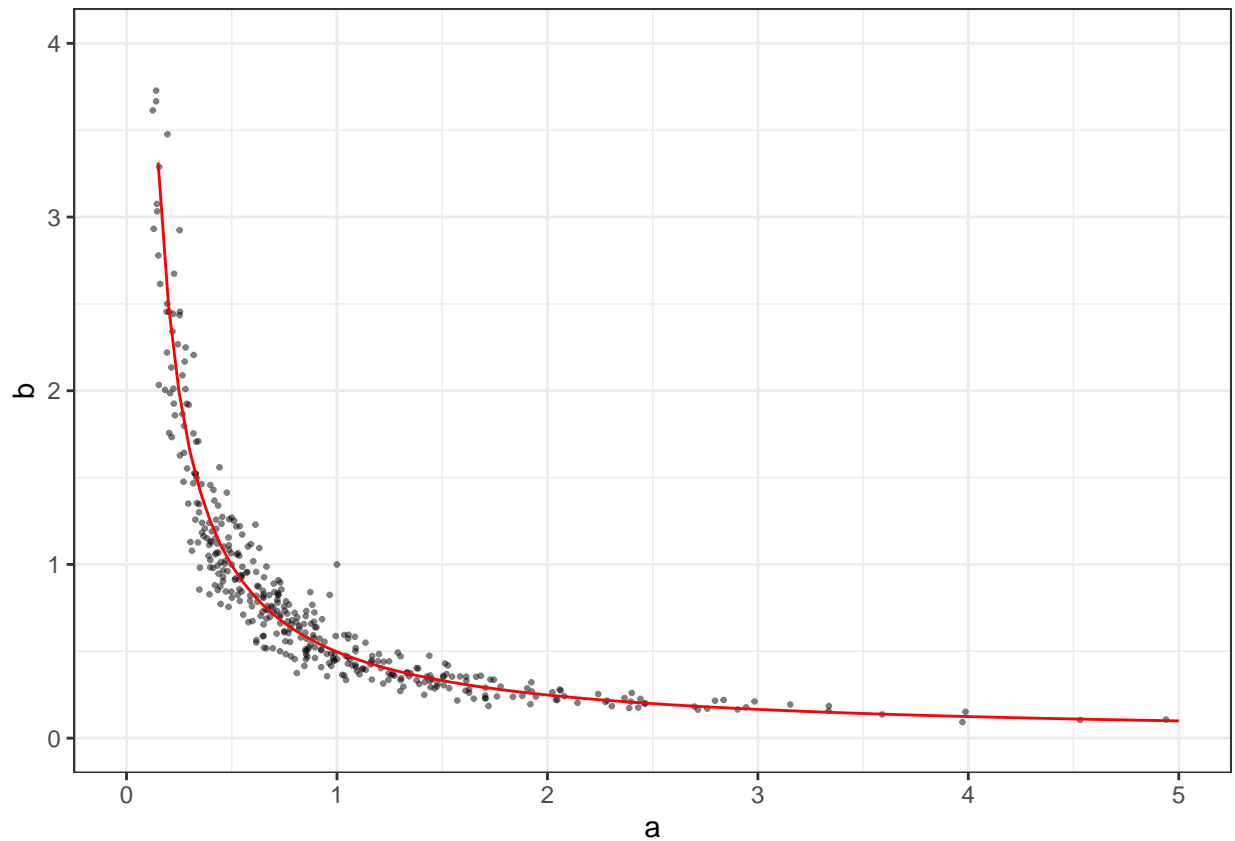
Plot a histogram or a density estimate of the estimated posterior using tasks (2) and (3).

```
prod=mean(ABthinned[,1]*ABthinned[,2]) # mean of a*b
f=function(x){return(prod/x)}
dft=data.frame(a=ABthinned[,1],b=ABthinned[,2])
```

```
ggplot(dft, aes(x = a, y = b)) +
  #geom_density_2d(aes(color = ..level..),bins=10)+
  geom_point(alpha=0.5,size=0.5)+
  xlim(0,5)+ylim(0,4)+
  geom_function(fun=f,color="red")+
  theme_bw()
```

```
## Warning: Removed 6 rows containing missing values ('geom_point()').
```



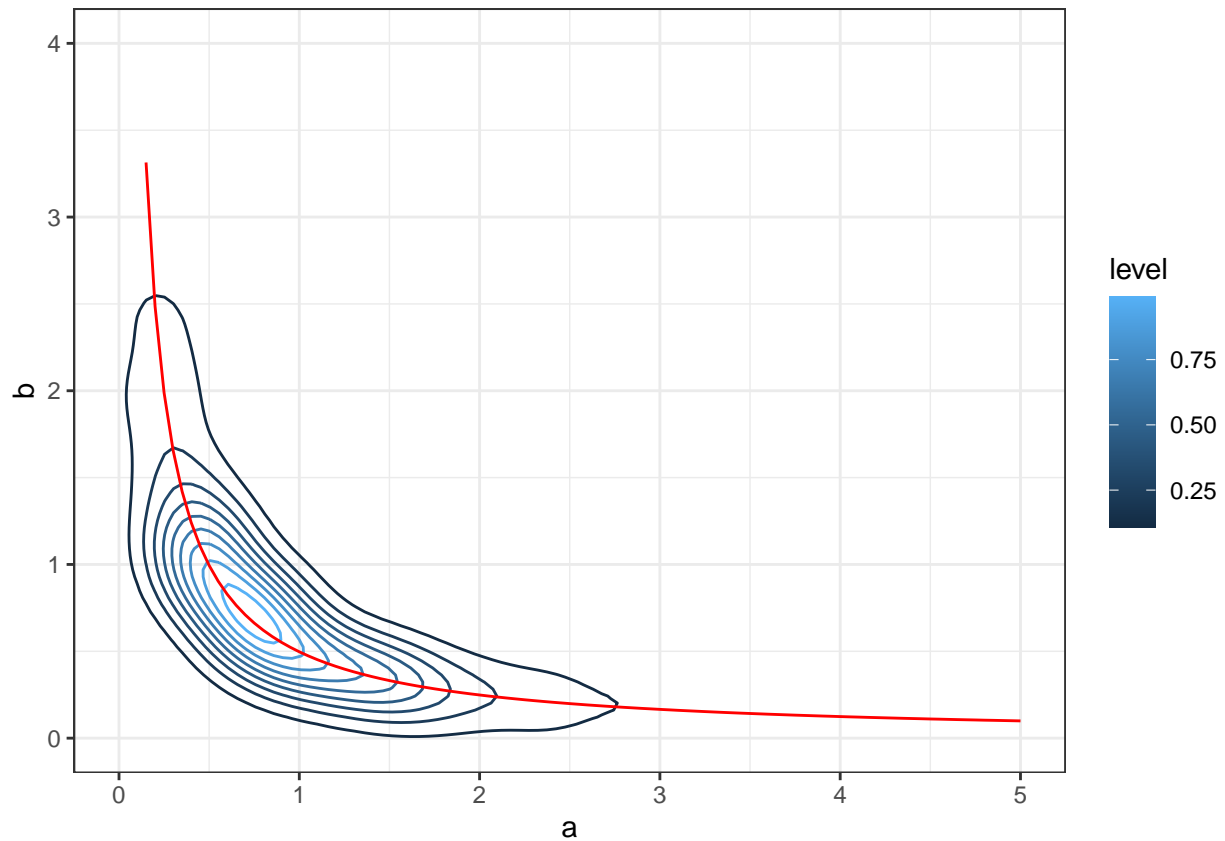


Plot the distribution density of  $a$  and  $b$  (higher density when  $a$  and  $b$  are similar and lower density when  $a$  and  $b$  are far from each other)

```
ggplot(dft, aes(x = a, y = b)) +
  geom_density_2d(aes(color = ..level..), bins=10)+
  #geom_point(alpha=0.1, size=0.5)+
  xlim(0,5)+ylim(0,4)+
  geom_function(fun=f, color="red")+
  theme_bw()
```

```
## Warning: The dot-dot notation ('..level..') was deprecated in ggplot2 3.4.0.
## i Please use 'after_stat(level)' instead.
```

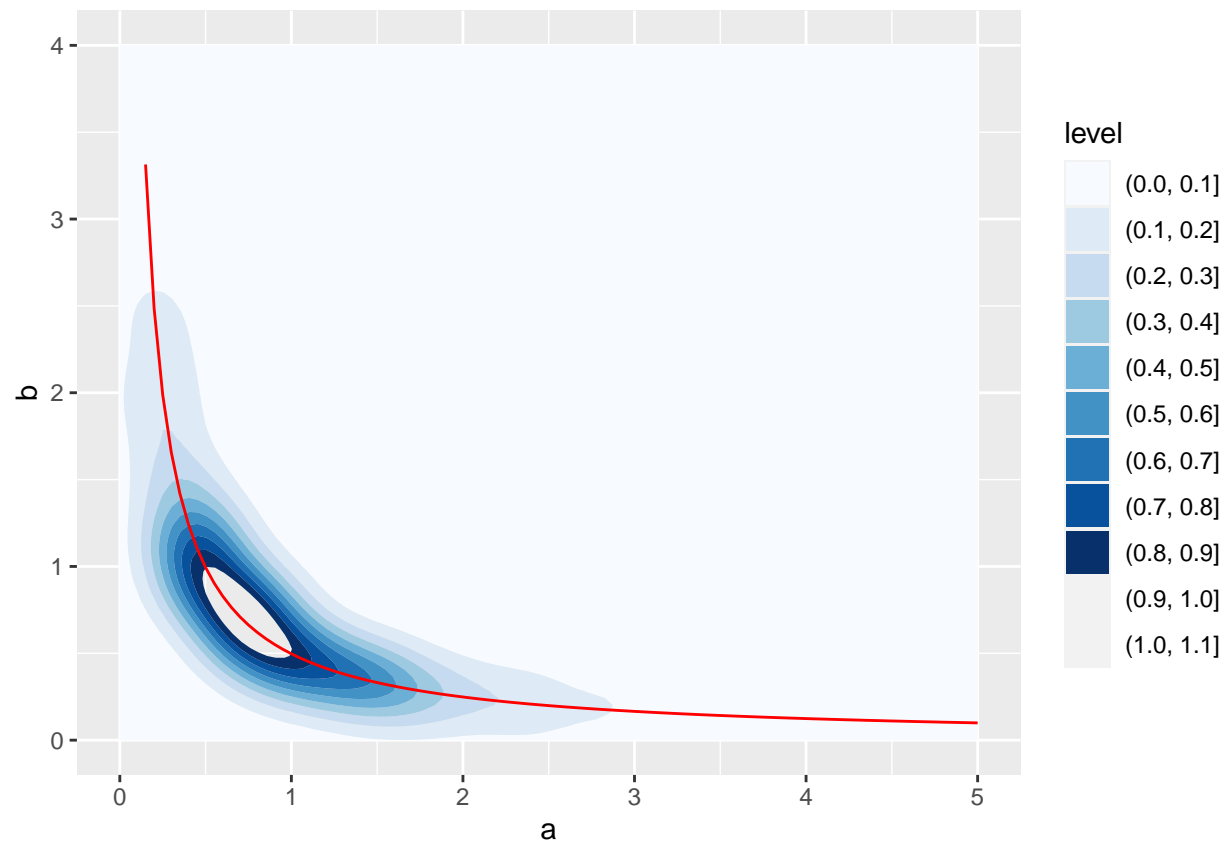
```
## Warning: Removed 6 rows containing non-finite values ('stat_density2d()').
```



```
ggplot(dft, aes(x = a, y = b)) +
  geom_density_2d_filled() +
  scale_fill_brewer()+
  xlim(0,5)+ylim(0,4)+
  geom_function(fun=f,color="red")
```

```
## Warning: Removed 6 rows containing non-finite values
## ('stat_density2d_filled()').
```

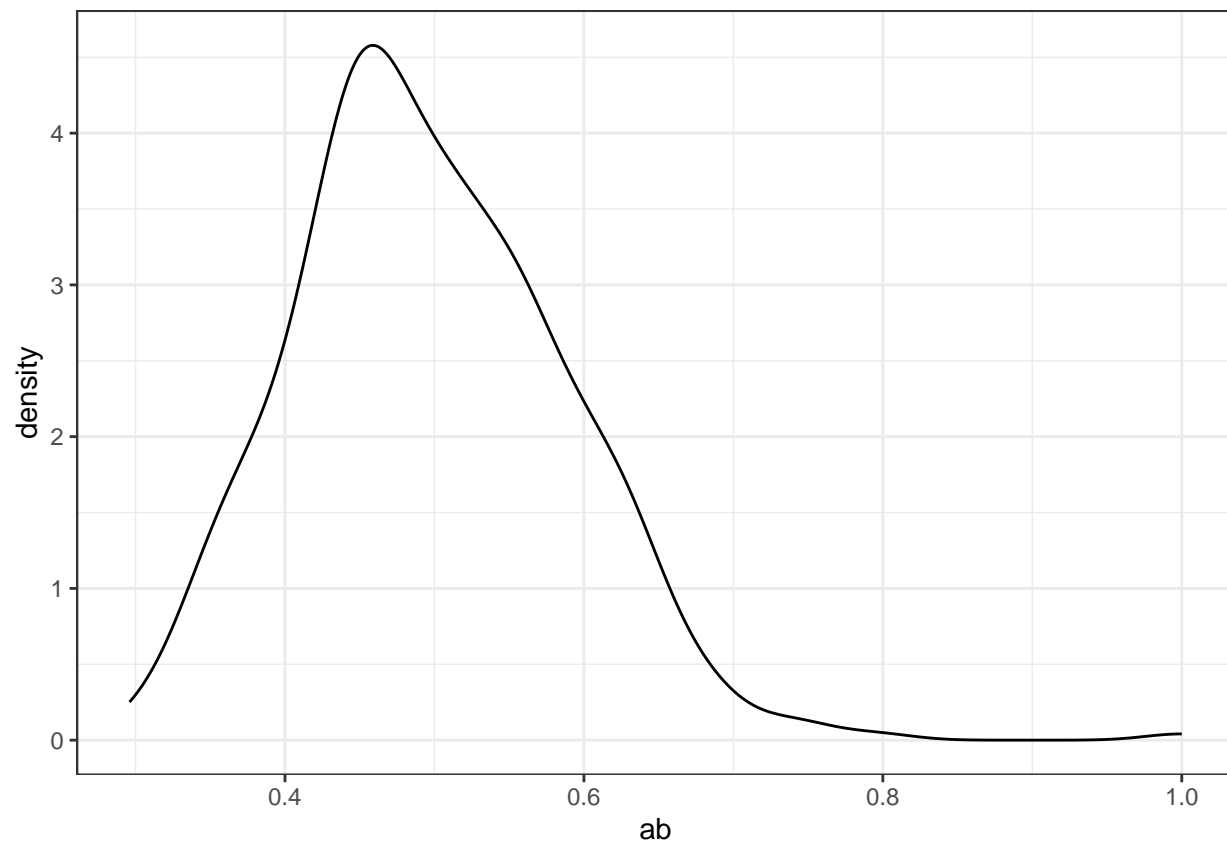
```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Blues is 9
## Returning the palette you asked for with that many colors
```



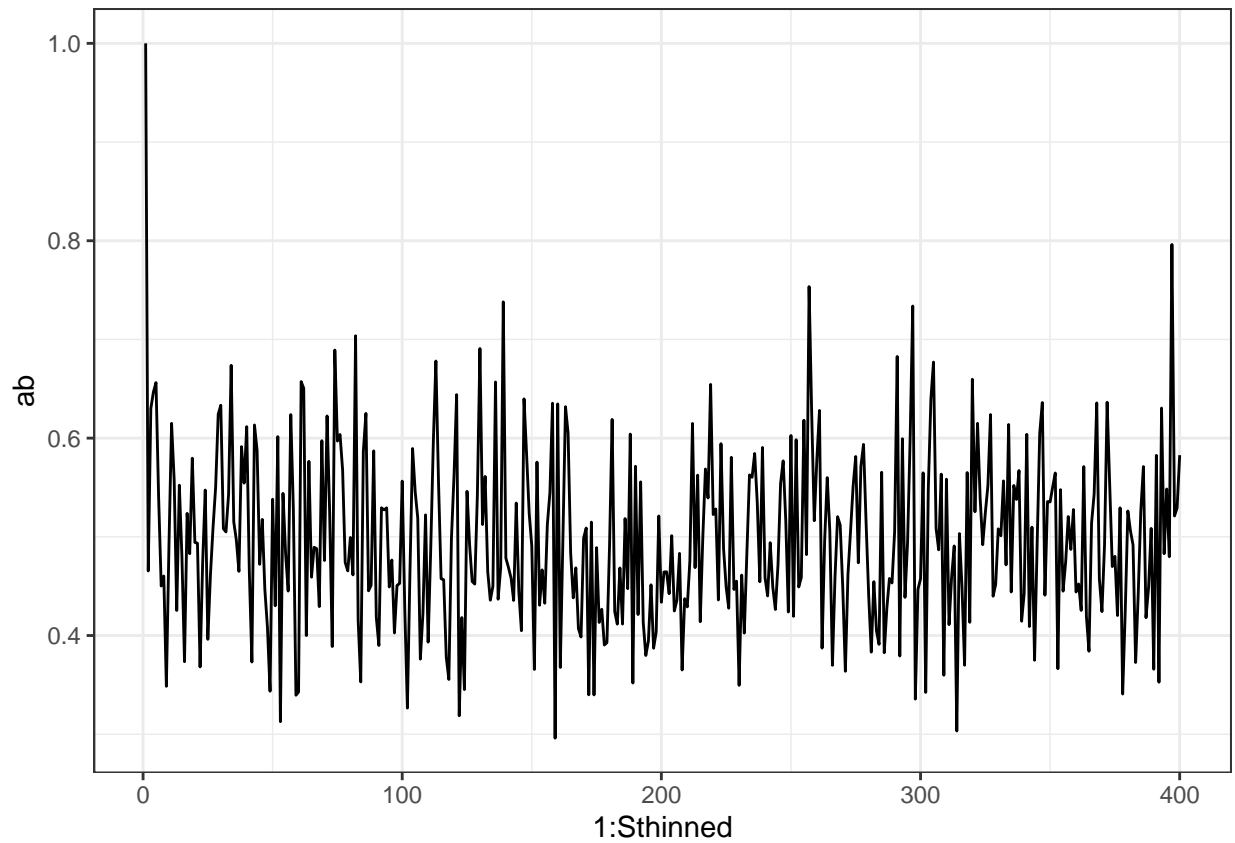
## Task 5

Why am I plotting this thin red line?

```
# the distribution of a*b  
ab=ABthinned[,1]*ABthinned[,2]  
ggplot()+  
  geom_density(aes(x=ab))+  
  theme_bw()
```



```
ggplot()+  
  geom_line(aes(x=1:Sthinned, y=ab))+  
  theme_bw()
```



From these plots it seems that  $ab$  can be estimated much more easily than  $a$  and  $b$ . Why is that? (**Identifiability**)

**Q: How do you know that your estimated posterior in task (3) is reliable?**

A: The density of  $a$  and  $b$  are very similar, so they have symmetric posterior density, which verifies our assumption before.